

A context dependent pair hidden Markov model for statistical alignment

Ana Arribas-Gil

Departamento de Estadística, Universidad Carlos III de Madrid, C/ Madrid, 126 - 28903 Getafe, Spain.
email: ana.arribas@uc3m.es

and

Catherine Matias

Laboratoire Statistique et Génome, Université d'Évry Val d'Essonne, UMR CNRS 8071, USC INRA,
 523 pl. des Terrasses de l'Agora, 91000 Évry, France.
email: catherine.matias@genopole.cnrs.fr

SUMMARY: This article proposes a novel approach to statistical alignment of nucleotide sequences by introducing a context dependent structure on the substitution process in the underlying evolutionary model. We propose to estimate alignments and context dependent mutation rates relying on the observation of two homologous sequences. The procedure is based on a generalized pair-hidden Markov structure, where conditional on the alignment path, the nucleotide sequences follow a Markov distribution. We use a stochastic approximation expectation maximization (**saem**) algorithm to give accurate estimators of parameters and alignments. We provide results both on simulated data and vertebrate genomes, which are known to have a high mutation rate from CG dinucleotide. In particular, we establish that the method improves the accuracy of the alignment of a human pseudogene and its functional gene.

KEY WORDS: Comparative genomics; Contextual alignment; DNA sequence alignment; EM algorithm; Insertion deletion model; Pair hidden Markov model; Probabilistic alignment; Sequence evolution; Statistical alignment; Stochastic expectation maximization algorithm.

1. Introduction

Alignment of DNA sequences is the core process in comparative genomics. An alignment is a mapping of the nucleotides in one sequence onto the nucleotides in the other sequence which captures two important evolutionary processes: the substitution (of a nucleotide, by another one) and the insertion or deletion (indel, of one or several nucleotides). Thus, in this mapping, two *matched* nucleotides are supposed to derive from a common ancestor (they are homologous) and the mapping also allows for *gaps* which may be introduced into one or the other sequence. Several powerful alignment algorithms have been developed to align two or more sequences, a vast majority of them relying on dynamic programming methods that optimize an alignment score function. While score-based alignment procedures appear to be very rapid (an important point regarding to the current sizes of the databases), they have at least two major flaws: i) The choice of the score parameters requires some biological expertise and is non objective. This is particularly problematic since the resulting alignments are not robust to this choice (Dewey et al., 2006). ii) The highest scoring alignment might not be the most relevant from a biological point of view. One should then examine the k -highest scoring alignments (with the additional issue of selecting a value for k) but this approach often proves impractical because of the sheer number of suboptimal alignments (Waterman and Eggert, 1987). We refer to Chapters 2 and 4 in Durbin et al. (1998) for an introduction to sequence alignment.

Contrarily to score-based alignment procedures, statistical (or probabilistic) alignment methods rely on an explicit modeling of the two evolutionary processes at the core of an alignment, namely the substitution and the indel processes. While there is a general agreement that substitution events are satisfactorily modeled relying on continuous time Markov processes on the nucleotide state space, very few indel models have been proposed in the literature and none of them has become a standard. The first evolutionary model for biological sequences dealing with indels was formulated by Thorne et al. (1991), hereafter the TKF91 model. It provides a basis for performing alignment within a statistical framework: the *optimal* alignment is obtained as the evolutionary path that maximizes the likelihood of observing the sequences as being derived from a common ancestor (under this model). This has at least two major advantages over the score-based method. First, in this context, the evolutionary parameters (which play the role of an underlying scoring function) are not chosen by the user but rather estimated from the data, in a maximum likelihood framework. Second, a posterior probability is obtained on the set of alignments: the analysis

does not concentrate on the highest probable path and rather provides sample alignments from this posterior distribution. Moreover, a *confidence measure* may be assigned to each position in the alignment. This is particularly relevant as alignment uncertainty is known to be a major flaw in comparative genomics (Wong et al., 2008).

Statistical alignment is performed under the convenient pair-hidden Markov model, hereafter pair-HMM (see Arribas-Gil et al., 2006; Durbin et al., 1998), enabling the use of generalizations of the expectation-maximization (**em**) algorithm (Dempster et al., 1977). Some drawbacks of the preliminary TKF91 proposal have first been improved by the same authors in what is called the TKF92 version of the model (Thorne et al., 1992). Then, the original models have been later refined in many ways, as for instance in Arribas-Gil et al. (2009); Knudsen and Miyamoto (2003); Metzler (2003); Miklós (2003); Miklós et al. (2004); Miklós and Toroczka (2001). We underline that an important characteristic of all those models is that the indel and the substitution processes are handled independently. An introduction to statistical alignment may be found in Lunter et al. (2005).

It is important to note that the original TKF91 model and all the later developments are primarily focused on modeling the indel process. As for the underlying substitution model, pair-HMMs procedures mostly rely on the simplest one: the one-parameter Jukes-Cantor model (Jukes and Cantor, 1969). Surprisingly, while models of substitution processes have made important progress towards a more accurate description of reality in the past years (see for instance Felsenstein, 2003, Chapter 13), these developments have had almost no impact in the statistical alignment literature. One of the reasons for that may be the existence of important biases found in sequence alignment and resulting from incorrectly inferred indels. For instance, Lunter et al. (2007) exhibit three major classes of such biases: (1) gap wander, resulting from the incorrect placement of gaps due to spurious non homologous similarity; (2) gap attraction, resulting in the joining of two closely positioned gaps into one larger gap; and (3) gap annihilation, resulting in the deletion of two indels of equal size for a typically more favorable representation as substitutions (see also Holmes and Durbin, 1998). This might explain a stronger focus on accurate estimation of indels positions. Another reason could be that since evolutionary parameters are to be estimated in statistical alignments procedures, their number should remain quite low.

On the contrary, score-based alignment methods offer some variety in the form of the match scoring functions, which is the counterpart of the substitution process parameters when relying on statistical alignment. Note that in this context, using different types of indels scores is hardly possible, since the dynamic programming approach limits those scores to affine (in the gap length) penalties. Nonetheless, the match scores may take different forms. One of the most recent improvements in score-based alignment procedures concerns the development of context-based scoring schemes as in Gambin et al. (2006); Gambin and Wojtawicz (2007) (see also Huang, 1994, as a much earlier attempt). In these works, the cost of a substitution depends on the surrounding nucleotides. As a consequence, the score of an alignment depends on the order of editing operations. Another drawback of this approach lies in the issue of computing a *p*-value associated to the alignment significance. Indeed, in order to assess the statistical significance of an alignment, one requires the knowledge of the tail distribution of the score statistic, under the null hypothesis of non homology between the sequences. While this distribution is not even fully characterized in the simpler case of site independent scoring schemes allowing gaps, it is currently out of reach when introducing dependencies in the substitution cost.

Nonetheless, allowing for context dependence in the modeling of the evolutionary processes is important as it captures local sequence dependent fluctuations in substitution rates. It is well-known for instance that, at least in vertebrates, there is a higher level of methylated cytosine in CG dinucleotides (denoted CpG). This methylated cytosine mutates abnormally frequently to thymine which results in higher substitution rates from the pair CpG (Bulmer, 1986). At the mutational level, context dependent substitution rates can have a significant impact on prediction accuracy (Siepel and Haussler, 2004).

Note that allowing for context dependent indel rates is also a challenging issue. For instance, microsatellite expansion and contraction is known to be a context dependent process (see for instance some recent results in Varela and Amos, 2009) and suggests the use of context dependent indel rates. Such an attempt has been very recently made by Hickey and Blanchette (2011a,b), relying on tree adjoining grammars. However, introducing locally dependent indel rates in an evolutionary process implies that the substitution and the indel processes may not anymore be handled separately. This independence between the two processes is a key assumption in limiting the computational cost of pair-HMMs and removing this hypothesis is beyond the scope of the present work. Thus, we limit our approach to modeling context dependent substitution rates.

In this work, we formulate a context dependent pair hidden Markov model, so as to perform statistical alignment of two sequences allowing for context dependent substitution rates. We then propose a method to perform parameter maximum likelihood estimation as well as posterior sampling within this model. Our procedure relies on stochastic versions (Celeux and Diebolt, 1985; Diebolt and Celeux, 1993) of the **em** algorithm. A similar method has already been successfully used in Arribas-Gil et al. (2009) in a classical pair-HMM (with no context dependent substitution rates) in which case it gives results very similar to Gibbs sampling. Note that Holmes (2005) also suggests the use of a stochastic version of **em** for estimating indel rates in the multiple alignment framework, assuming the alignment is unknown. In the same spirit, Hobolth (2008) proposes an **MCMC-EM** algorithm for the estimation of neighbor-dependent substitution rates from a multiple alignment and a phylogenetic tree. The advantages of stochastic versions of **em** algorithm is that they allow to perform maximum likelihood estimation in a hidden variables framework where the maximization is not explicit, the computation of the likelihood is computationally very expensive, and where the use of the classical **em** algorithm is not feasible. To our knowledge, this work is, with Hickey and Blanchette (2011a)'s approach, one of the first attempts to perform a contextual

statistical alignment.

This article is organized as follows. In Section 2, we first describe a context dependent pair hidden Markov model. In this framework, parameter maximum likelihood estimation combined with hidden states posterior sampling provides (a posterior probability on) contextual alignments of two sequences which is data-driven in the sense that it does not require any parameter choice. Section 3 briefly discusses asymptotic consistency results that can be obtained on the parameter estimation procedure. Then, Section 4 describes two versions of the **em** algorithm applied in this framework of contextual statistical alignment: the stochastic expectation maximization (**sem**) and the stochastic approximation expectation maximization (**saem**) (Delyon et al., 1999) algorithms. The performance of these procedures are illustrated on synthetic data in Section 5 and a real data set is handled in Section 6.

2. Model

A pair-hidden Markov model is described by the distribution of a non-observed increasing path through the two-dimensional integer lattice $\mathbb{N} \times \mathbb{N}$ and given this hidden path, the conditional distribution of a pair of observed sequences $X_{1:n} := X_1, \dots, X_n \in \mathcal{A}^n$ and $Y_{1:m} := Y_1, \dots, Y_m \in \mathcal{A}^m$, with values on a finite set \mathcal{A} (the nucleotides alphabet). The path reflects the insertion-deletion process acting on the sequences and corresponds to the *bare* alignment, namely the alignment without specification of the nucleotides. It is the sum of random moves from the set $\mathcal{E} = \{(1, 1), (1, 0), (0, 1)\}$ also denoted $\{M, I_X, I_Y\}$, where M stands for 'match', I_X stands for 'insert in X ' and similarly I_Y stands for 'insert in Y '. This path represents an empty alignment of the sequences, as the move M represents an homologous site between the two sequences, the move I_X represents either an insertion in the first sequence ($X_{1:n}$) or a deletion in the second ($Y_{1:m}$) and similarly for the move I_Y (see Figure 1).

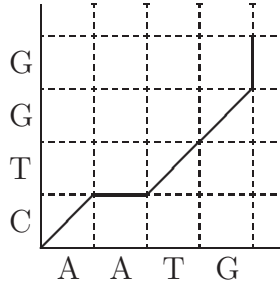


Figure 1. Graphical representation of an alignment between two sequences $X = AATG$ and $Y = CTGG$. The displayed alignment is $\begin{smallmatrix} & A & A & T & G & - \\ C & - & T & G & G & \end{smallmatrix}$.

We assume that the hidden path follows a Markov distribution without specifying from which insertion-deletion process it comes from. This could be for instance the TKF91 insertion-deletion process (Thorne et al., 1991) or any of its later variants (Arribas-Gil et al., 2009; Knudsen and Miyamoto, 2003; Metzler, 2003; Miklós, 2003; Thorne et al., 1992). Given the empty alignment, we then describe the conditional distribution of the sequence letters. In a 'match' position, this distribution reflects the substitution process while in 'insert in X ' or 'insert in Y ' positions, this distribution reflects both the substitution process and the distribution of inserted segments.

More precisely, let $\{\varepsilon_t\}_{t \geq 0}$ be an ergodic, homogeneous and stationary Markov chain with state space $\mathcal{E} = \{(1, 1), (1, 0), (0, 1)\}$ and transition probability π (a stochastic matrix of size 3×3). For notational convenience, the state space \mathcal{E} is sometimes equivalently denoted by $\{M, I_X, I_Y\}$. In this context, we might use obvious notations as for instance π_{MM} for the conditional probability $\mathbb{P}(\varepsilon_t = (1, 1) | \varepsilon_{t-1} = (1, 1))$.

Let $Z_t = (N_t, M_t) = \sum_{s=1}^t \varepsilon_s$ be the path induced by this Markov chain through the lattice $\mathbb{N} \times \mathbb{N}$. In the following, a path through the lattice $\mathbb{N} \times \mathbb{N}$ is always assumed to be increasing, namely of the form $\sum_s \varepsilon_s$ for some sequence $e \in \mathcal{E}^{\mathbb{N}}$. We denote by \mathbb{P}_π the corresponding stationary probability and by $\mathcal{E}_{n,m}$ the set of paths in $\mathbb{N} \times \mathbb{N}$ starting at $(0, 0)$ and ending at (n, m) . For any $e \in \mathcal{E}_{n,m}$, we also let $|e|$ denote the length of e , which satisfies $n \vee m \leq |e| \leq n + m$.

The hidden process generates the observed sequences in the following way. Given the hidden path $\{\varepsilon_t\}_{t \geq 0}$, the sequences $\{X_{1:n}, Y_{1:m}\}$ are generated according to an order one Markov chain: for any $e \in \mathcal{E}_{n,m}$,

$$\mathbb{P}(X_{1:n}, Y_{1:m} | \varepsilon = e) = \prod_{s=1}^{|e|} \mathbb{P}(X_{N_s}, Y_{M_s} | e_s, e_{s-1}, X_{N_{s-1}}, Y_{M_{s-1}}).$$

Note that the indexes N_s, M_s of the observations generated at time s are random. This is in sharp contrast with the classical hidden Markov model (HMM) and is specific to pair-HMMs (see Arribas-Gil et al., 2006, for more details).

In this work, we restrict our attention to order-one Markov conditional distributions. Note that the following developments could be generalized to higher order Markov chains at the cost of increased computational burden.

As we want to model the dependency in the substitution process only, we further constrain the model so that the dependency occurs only in successive homologous (match) positions. We thus use the following parametrization

$$\mathbb{P}(X_{N_s}, Y_{M_s} | e_s, e_{s-1}, X_{N_{s-1}}, Y_{M_{s-1}}) = \begin{cases} \tilde{h}(X_{N_s}, Y_{M_s} | X_{N_{s-1}}, Y_{M_{s-1}}) & \text{if } e_s = e_{s-1} = (1, 1), \\ h(X_{N_s}, Y_{M_s}) & \text{if } e_s = (1, 1) \neq e_{s-1}, \\ f(X_{N_s}) & \text{if } e_s = (1, 0), \\ g(Y_{M_s}) & \text{if } e_s = (0, 1), \end{cases}$$

where f, g are probability measures (p.m.) on \mathcal{A} ; function h is a p.m. on \mathcal{A}^2 and \tilde{h} may be viewed as a stochastic matrix on \mathcal{A}^2 .

Obvious necessary conditions for the parameters to be identifiable in this model are the following

$$\begin{cases} i) & \exists a, b \in \mathcal{A} \text{ such that } h(a, b) \neq f(a)g(b), \\ ii) & \exists a, b, c, d \in \mathcal{A} \text{ such that } \tilde{h}(a, b|c, d) \neq f(a)g(b), \\ iii) & \exists a, b, c, d \in \mathcal{A} \text{ such that } \tilde{h}(a, b|c, d) \neq h(a, b). \end{cases} \quad (1)$$

The whole parameter set is then given by

$$\Theta = \{\theta = (\pi, f, g, h, \tilde{h}); \theta \text{ satisfies (1)}\}.$$

Statistical alignment of two sequences consists in maximizing with respect to θ the following criterion $w_{n,m}(\theta)$ (Durbin et al., 1998). This criterion plays the role of a log-likelihood in the pair-HMM. (For a discussion on the quantities playing the role of likelihoods in pair-HMMs, see Arribas-Gil et al., 2006). For any integers $n, m \geq 1$ and any observations $X_{1:n}$ and $Y_{1:m}$, let

$$w_{n,m}(\theta) := \log Q_\theta(X_{1:n}, Y_{1:m}) := \log \mathbb{P}_\theta(\exists s \geq 1, Z_s = (n, m); X_{1:n}, Y_{1:m}). \quad (2)$$

The criterion $w_{n,m}$ is more explicitly defined as

$$w_{n,m}(\theta) = \log \left(\sum_{s=n \vee m}^{n+m} \sum_{e \in \mathcal{E}_{n,m}, |e|=s} \mathbb{P}_\pi(\varepsilon_{1:s} = e_{1:s}) \times \left\{ \prod_{k=1}^s f(X_{n_k})^{1\{e_k=(1,0)\}} g(Y_{m_k})^{1\{e_k=(0,1)\}} \right. \right. \\ \left. \left. \times h(X_{n_k}, Y_{m_k})^{1\{e_k=(1,1), e_{k-1} \neq (1,1)\}} \tilde{h}(X_{n_k}, Y_{m_k} | X_{n_{k-1}}, Y_{m_{k-1}})^{1\{e_k=e_{k-1}=(1,1)\}} \right\} \right),$$

where $(n_k, m_k) = \sum_{t=1}^k e_t$. Now, we define the parameter estimator as

$$\hat{\theta}_{n,m} = \arg \max_{\theta \in \Theta} w_{n,m}(\theta).$$

In the classical pair-HMM (namely without accounting for dependencies in the substitution process), consistency results for the above estimator $\hat{\theta}_{n,m}$ were obtained in Arribas-Gil et al. (2006) and are extended here to the context dependent case in Section 3. Then in Section 4, we develop an **saem** algorithm to compute $\hat{\theta}_{n,m}$ and provide an a posteriori probability distribution over the set of alignments. Indeed, a main issue is to obtain an accurate alignment of the sequences. Score-based alignment methods heavily rely on the Viterbi algorithm for this purpose (see Durbin et al., 1998), which provides the most probable *a posteriori* alignment. However, optimal alignments often look different from typical ones, and providing a unique alignment as the result of the estimation procedure may not be very informative. For this reason, it may be more interesting to provide the probability distribution of hidden states over each pair of nucleotides from the observed sequences (for the parameter value $\hat{\theta}_{n,m}$). This can be done under the pair-HMM framework and gives us a reliability measure of any alignment of the two sequences (see Arribas-Gil et al., 2009).

3. Consistency results for parameter estimation

In this part, we give generalizations of results first obtained in Arribas-Gil et al. (2006), to the context dependent pair-HMM. As the proofs follow the same lines as in this reference, we shall omit them. We first introduce some notations and formulate some assumptions that will be needed. For any $\delta > 0$, we let

$$\Theta_\delta = \{\theta \in \Theta; \forall k, \theta_k \geq \delta\} \quad \text{and} \quad \Theta_0 = \{\theta \in \Theta; \forall k, \theta_k > 0\}.$$

The true parameter value will be denoted by θ_0 and is assumed to belong to the set Θ_0 . Probabilities and expectations under this parameter value are denoted by \mathbb{P}_0 and \mathbb{E}_0 , respectively. We now introduce a notation for the marginal distributions of

h and \tilde{h} . For any $b, c \in \mathcal{A}$, let

$$\begin{aligned} h_X(\cdot) &:= \sum_{a \in \mathcal{A}} h(\cdot, a), \quad h_Y(\cdot) := \sum_{a \in \mathcal{A}} h(a, \cdot), \\ \tilde{h}_X(\cdot|b, c) &:= \sum_{a \in \mathcal{A}} \tilde{h}(\cdot, a|b, c), \quad \tilde{h}_Y(\cdot|b, c) := \sum_{a \in \mathcal{A}} \tilde{h}(a, \cdot|b, c). \end{aligned}$$

We then let Θ_{marg} be a subset of parameters in Θ_0 satisfying some assumptions on the marginals of h and \tilde{h} .

$$\Theta_{\text{marg}} = \left\{ \theta \in \Theta_0; \forall b, c \in \mathcal{A}, h_X = f, h_Y = g, \tilde{h}_X(\cdot|b, c) = f(\cdot), \tilde{h}_Y(\cdot|b, c) = g(\cdot) \right\}.$$

Note that for instance, the parametrization used in Section 5 satisfies $\theta \in \Theta_{\text{marg}}$. Finally, we introduce the process w_t defined in a similar way as in (2) by

$$w_t(\theta) := \log Q_\theta(X_{1:N_t}, Y_{1:M_t}).$$

As previously noted, the length t of an alignment of sequences with respective sizes n and m satisfies $n \vee m \leq t \leq n + m$. Thus, asymptotic results for $t \rightarrow +\infty$ will imply equivalent ones for $n, m \rightarrow +\infty$. In other words, consistency results obtained when $t \rightarrow +\infty$ are valid for long enough observed sequences, even if one does not know the length t of the true underlying alignment. We now state the main theorem.

THEOREM 1: *For any $\theta \in \Theta_0$, we have*

i) *Renormalized log-likelihood $t^{-1}w_t(\theta)$ converges \mathbb{P}_0 -almost surely and in \mathbb{L}_1 , as t tends to infinity, to*

$$w(\theta) = \lim_{t \rightarrow \infty} \frac{1}{t} \mathbb{E}_0(\log Q_\theta(X_{1:N_t}, Y_{1:M_t})) = \sup_t \frac{1}{t} \mathbb{E}_0(\log Q_\theta(X_{1:N_t}, Y_{1:M_t})).$$

ii) *Moreover, $w(\theta_0) \geq w(\theta)$ and strict inequality is valid as long as at least one of these conditions is satisfied*

- A. $\theta \in \Theta_0$ and $\forall \lambda > 0, \mathbb{E}_\theta(\epsilon_1) \neq \lambda \mathbb{E}_0(\epsilon_1)$,
- B. $\theta_0, \theta \in \Theta_{\text{marg}}$ and either $f \neq f_0$ or $g \neq g_0$.

iii) *The family of functions $\{t^{-1}w_t(\theta)\}_{t \geq 1}$ is uniformly equicontinuous on the set Θ_δ , for any $\delta > 0$.*

The three statements in Theorem 1 are the key ingredients to establish the consistency of $\hat{\theta}_{n,m}$, following Wald's classical proof (Wald, 1949) of maximum likelihood estimators consistency (see also van der Vaart, 1998, Chapter 5). The most subtle part is point ii), that states some sufficient conditions under which the maximum of the limiting function w is attained only at the true parameter value θ_0 . For instance, condition A means that the main directions of the paths $\{Z_t\}_{t \geq 1}$ generated under the two parameter values θ and θ_0 are different. As a consequence, these two parameters may be easily distinguished from each other relying on the distributions \mathbb{P}_θ and \mathbb{P}_0 . Note that these two different cases (condition A or B satisfied) are certainly not exhaustive and we refer to Arribas-Gil et al. (2006) for simulation results indicating that point ii) might be true under other scenarios. Note in particular that for evolutionary models assuming that insertions and deletions happen at the same rate, $\mathbb{E}_\theta(\epsilon_1)$ is proportional to $(1, 1)$ for any parameter value θ . In this case, condition A) is never satisfied and we are not able to prove that $h \neq h_0$ or $\tilde{h} \neq \tilde{h}_0$ implies strict inequality $w(\theta_0) > w(\theta)$. This is due to the dependency structure on the sequences, that make it difficult to link the difference $w(\theta_0) - w(\theta)$ with a Kullback-Leibler divergence between densities h and h_0 or between \tilde{h} and \tilde{h}_0 (see the proof of Theorem 2 in Arribas-Gil et al., 2006, for more details).

In general, one is not interested in the full (context) pair-HMM parameters but rather in a sub-parametrization induced by the choice of a specific evolutionary model (see for instance the sub-parametrization proposed in Section 5). Let $\beta \mapsto \theta(\beta)$ be a continuous parametrization from some set B to Θ . For any $\delta > 0$, let $B_\delta = \theta^{-1}(\Theta_\delta)$. We assume that $\beta_0 := \theta^{-1}(\theta_0)$ belongs to B_δ for some $\delta > 0$. We then let

$$\hat{\beta}_{n,m} := \arg \max_{\beta \in B_\delta} w_{n,m}(\theta(\beta)).$$

Besides from giving results on the frequentist estimator $\hat{\beta}_{n,m}$, we consider Bayesian estimates of the parameters. Let ν be a prior probability measure on the set B_δ . Markov chain Monte Carlo (MCMC) algorithms approximate the random distribution $\nu_{n,m}$, interpreted as the posterior measure given observations $X_{1:n}$ and $Y_{1:m}$ and defined by

$$\nu_{n,m}(d\beta) = \frac{Q_{\theta(\beta)}(X_{1:n}, Y_{1:m})\nu(d\beta)}{\int_{B_\delta} Q_{\theta(\beta')}(X_{1:n}, Y_{1:m})\nu(d\beta')}.$$

With these definitions at hand, we can now state the following corollary.

COROLLARY 1: *If the maximizer of $\beta \rightarrow w(\theta(\beta))$ over the set B_δ is reduced to the singleton $\{\beta_0\}$, then we have the following results*

i) $\hat{\beta}_{n,m}$ *converges \mathbb{P}_0 -almost-surely to β_0 ,*

ii) If ν puts some weight on a neighborhood of β_0 , then the sequence of posterior measures $\nu_{n,m}$ converges in distribution, \mathbb{P}_0 -almost surely, to the Dirac mass at β_0 .

4. Algorithms

The approach we use here to maximize criterion (2) and provide a posterior distribution on the set of alignments, given two observed sequences relies on a stochastic version (Celeux and Diebolt, 1985; Diebolt and Celeux, 1993) of **em** algorithm (Baum et al., 1970; Dempster et al., 1977). This approach has already been successfully used in Arribas-Gil et al. (2009) in a classical pair-HMM (meaning with no context dependent substitution rates) in which case it gives results very similar to Gibbs sampling. It allows to perform maximum likelihood estimation in a framework in which the maximization is not explicit, the computation of the likelihood is computationally very expensive, and in which the use of **em** algorithm is not feasible (see Appendix for details).

4.1 Forward and backward equations

We describe below the forward and backward probabilities used in the later procedures. The formulas are easily obtained by generalizing those from the classical pair-HMM (Arribas-Gil, 2007; Durbin et al., 1998).

The forward equations are obtained as follows. For any value $u \in \mathcal{E}$, let $\mathcal{E}_{n,m}^u = \{e \in \mathcal{E}_{n,m}, e|_{[e]} = u\}$ be the set of paths ending at (n, m) with last step being u . Then the forward probabilities are defined for any $i \leq n$ and $j \leq m$, as

$$\alpha^u(i, j) = \mathbb{P}_\theta(\exists s \geq 1, Z_s = (i, j), \varepsilon_s = u, X_{1:i}, Y_{1:j}) = \sum_{e \in \mathcal{E}_{n,m}^u} \mathbb{P}_\theta(\varepsilon = e, X_{1:i}, Y_{1:j}).$$

These forward probabilities are computed recursively in the following way. We first set the initial values

$$\begin{aligned} \forall u \in \mathcal{E}, \quad \alpha^u(0, -1) &= \alpha^u(-1, 0) = 0 \\ \forall u \in \{I_X, I_Y\}, \quad \alpha^u(0, 0) &= 0 \text{ and } \alpha^M(0, 0) = 1. \end{aligned}$$

Then for $i = 0, \dots, n$ and $j = 0, \dots, m$ except $(i, j) = (0, 0)$, we recursively compute

$$\begin{aligned} \alpha^M(i, j) &= \mathbb{P}_\theta(\exists s \geq 1, \varepsilon_s = (1, 1), Z_s = (i, j), X_{1:i}, Y_{1:j}) = \sum_{u \in \mathcal{E}} \mathbb{P}_\theta(\exists s \geq 1, \varepsilon_s = (1, 1), Z_s = (i, j), \varepsilon_{s-1} = u, X_{1:i}, Y_{1:j}) \\ &= \tilde{h}(X_i, Y_j | X_{i-1}, Y_{j-1}) \pi_{MM} \alpha^M(i-1, j-1) + h(X_i, Y_j) \pi_{IX} \alpha^{I_X}(i-1, j-1) + h(X_i, Y_j) \pi_{IY} \alpha^{I_Y}(i-1, j-1), \end{aligned} \quad (3)$$

and in the same way

$$\begin{aligned} \alpha^{I_X}(i, j) &= f(X_i) \sum_{u \in \mathcal{E}} \pi_{uI_X} \alpha^u(i-1, j), \\ \alpha^{I_Y}(i, j) &= g(Y_j) \sum_{u \in \mathcal{E}} \pi_{uI_Y} \alpha^u(i, j-1). \end{aligned}$$

Note that the log-likelihood of the observed sequences is then simply obtained from the forward probabilities by taking

$$w_{n,m}(\theta) = \log \mathbb{P}_\theta(\exists s \geq 1, Z_s = (n, m), X_{1:n}, Y_{1:m}) = \log(\alpha^{I_X}(n, m) + \alpha^{I_Y}(n, m) + \alpha^M(n, m)).$$

Note that the forward equations could thus be used to compute the log-likelihood of two observed sequences and a numerical optimization of this quantity could give maximum likelihood estimates. However, such a strategy fails short as soon as the dimension of the parameter space is not small. Indeed, the computational cost of calculating the log-likelihood for a given parameter value is $O(nm)$, and thus the computational cost of its maximization over a grid is $O(p^k nm)$, where k is the number of parameters and p the number of values considered for each parameter.

We now describe the backward probabilities and their recursive computation. For any value $u \in \mathcal{E}$, the backward probabilities are defined as

$$\beta^u(i, j) = \mathbb{P}_\theta(X_{i+1:n}, Y_{j+1:m} | \exists s \geq 1, Z_s = (i, j), \varepsilon_s = u, X_i, Y_j).$$

We initialize these values by taking, for any $u \in \mathcal{E}$,

$$\beta^u(n, m) = 1 \text{ and } \beta^u(n, m+1) = \beta^u(n+1, m) = 0,$$

and recursively compute, for $i = n, \dots, 1$ and $j = m, \dots, 1$, except (n, m) , the quantities

$$\begin{aligned} \beta^u(i, j) &= \pi_{uM} \beta^M(i+1, j+1) \{h(X_{i+1}, Y_{j+1}) 1_{u \in \{I_X, I_Y\}} + \tilde{h}(X_{i+1}, Y_{j+1} | X_i, Y_j) 1_{u=M}\} \\ &\quad + \pi_{uI_X} \beta^{I_X}(i+1, j) f(X_{i+1}) + \pi_{uI_Y} \beta^{I_Y}(i, j+1) g(Y_{j+1}). \end{aligned} \quad (4)$$

Note that contrarily to the case of HMM, the forward-backward probabilities do not give rise to the classical **em** strategy, as we do not obtain from these equations the conditional expectation of the complete log-likelihood, given the two observed sequences (see Appendix for further details).

4.2 sem and saem algorithms

The forward-backward probabilities may be used to simulate the hidden path, conditional on the observed sequences. Thus, **em** algorithm may be replaced by **sem** (Celeux and Diebolt, 1985) or **saem** (Delyon et al., 1999) procedures. Indeed, let us explain the basic idea of these stochastic versions of **em** algorithm. Denoting by L_{nm} the random value $s \geq 1$ such that $Z_s = (n, m)$ (the first and only hitting time for the point (n, m) , which is not necessarily finite), the complete log-likelihood is

$$\log \mathbb{P}_\theta(X_{1:n}, Y_{1:m}, L_{nm}, \varepsilon_{1:L_{nm}})$$

(see Appendix for details) and for a current parameter value θ' , its conditional expectation writes

$$Q_{\theta'}(\theta) := \mathbb{E}_{\theta'}(\log \mathbb{P}_\theta(X_{1:n}, Y_{1:m}, L_{nm}, \varepsilon_{1:L_{nm}}) | X_{1:n}, Y_{1:m}).$$

Now, the idea of these stochastic approximations is to replace the computation of $Q_{\theta'}(\theta)$ by an approximation obtained from the simulation (under parameter value θ') of a number of hidden paths $\{\varepsilon_{1:s}\}$ in $\mathcal{E}_{n,m}$, with possibly different lengths. These paths satisfy the property that for each obtained value of s , the paths with length s have the same occurrence probability as $\mathbb{P}_{\theta'}(\varepsilon | X_{1:n}, Y_{1:m}, L_{nm} = s)$. Note that the length of each simulated path is random. Then the maximization with respect to θ of the complete log-likelihood is done on the basis of the simulated alignments. As a consequence, this maximization is performed via a simple counting of specific events and does not require numerical procedures.

In particular, iteration r of **sem** algorithm writes

- *Simulation step*: generate one hidden path e^r of some random length s with the same distribution as

$$\mathbb{P}_{\theta^{(r-1)}}(\varepsilon_{1:s} | X_{1:n}, Y_{1:m}, L_{nm} = s).$$

- *Maximization step*: $\theta^{(r)} = \arg \max_{\theta \in \Theta} \log \mathbb{P}_\theta(X_{1:n}, Y_{1:m}, \varepsilon = e^r)$.

And iteration r of **saem** algorithm writes

- *Simulation step*: generate $m(r)$ hidden paths $e^r(j)$, $j = 1, \dots, m(r)$ with resulting random lengths $s_{r,j}$, each one having the same distribution as

$$\mathbb{P}_{\theta^{(r-1)}}(\varepsilon_{1:s_{r,j}} | X_{1:n}, Y_{1:m}, L_{nm} = s_{r,j}).$$

- *Stochastic approximation step*: update

$$Q_r(\theta) = Q_{r-1}(\theta) + \gamma_r \left(\frac{1}{m(r)} \sum_{j=1}^{m(r)} \log \mathbb{P}_\theta(X_{1:n}, Y_{1:m}, \varepsilon = e^r(j)) - Q_{r-1}(\theta) \right),$$

where $\{\gamma_r\}_{r \geq 1}$ is a decreasing sequence of positive step size and $Q_0(\theta) = 1/m(0) \sum_{j=1}^{m(0)} \log \mathbb{P}_\theta(X_{1:n}, Y_{1:m}, \varepsilon = e^0(j))$.

- *Maximization step*: $\theta^{(r)} = \arg \max_{\theta \in \Theta} Q_r(\theta)$.

The simulation step is common to both algorithms and consists in drawing paths of some unknown length s with same distribution as $\mathbb{P}_{\theta^{(r-1)}}(\varepsilon_{1:s} | X_{1:n}, Y_{1:m}, L_{nm} = s)$. This is performed via the backwards sampling based on the forward probabilities as we now explain. We may write

$$\begin{aligned} \mathbb{P}_{\theta^{(r-1)}}(\varepsilon_{1:s} | X_{1:n}, Y_{1:m}, L_{nm} = s) &= \mathbb{P}_{\theta^{(r-1)}}(\varepsilon_s | X_{1:n}, Y_{1:m}, L_{nm} = s) \prod_{k=1}^{s-1} \mathbb{P}_{\theta^{(r-1)}}(\varepsilon_k | \varepsilon_{k+1:s}, X_{1:n}, Y_{1:m}, L_{nm} = s) \\ &= \mathbb{P}_{\theta^{(r-1)}}(\varepsilon_s | X_{1:n}, Y_{1:m}, Z_s = (n, m)) \prod_{k=1}^{s-1} \mathbb{P}_{\theta^{(r-1)}}(\varepsilon_k | \varepsilon_{k+1}, Z_{k+1}, X_{1:N_{k+1}}, Y_{1:M_{k+1}}). \end{aligned} \quad (5)$$

The last equality comes from the following: Conditional on ε_{k+1} , the random variable ε_k does not anymore depend on the observations after time $k+2$. If we moreover condition on the variables $\varepsilon_{k+1:s}$ and $L_{nm} = s$, then we know which point (i, j) on the lattice corresponds to $Z_{k+1} = (n, m) - \sum_{t=k+2}^s \varepsilon_t = (i, j)$. The values of the observed sequence after time $k+2$ correspond exactly to $X_{i+1:n}, Y_{j+1:m}$. Then conditional on $\{\varepsilon_{k+1:s}, X_{1:n}, Y_{1:m}, L_{nm} = s\}$, the variable ε_k only depends on ε_{k+1} (according to the Markov property) and on the values $Z_{k+1}, X_{1:N_{k+1}}, Y_{1:M_{k+1}}$.

So, given ε_{k+1} , the position $Z_{k+1} = (i, j)$ on the lattice and the observed values $X_{1:i}, Y_{1:j}$, we want to sample ε_k from $\mathbb{P}_{\theta^{(r-1)}}(\varepsilon_k | \varepsilon_{k+1}, X_{1:i}, Y_{1:j}, Z_{k+1} = (i, j))$, for $k = s, s-1, \dots, 1$. Up to a renormalizing constant, each step corresponds to sampling from $\mathbb{P}_{\theta^{(r-1)}}(\varepsilon_k, \varepsilon_{k+1} | X_{1:i}, Y_{1:j}, Z_{k+1} = (i, j))$ or equivalently from $\pi_{\varepsilon_k, \varepsilon_{k+1}} \alpha^{\varepsilon_k}((i, j) - \varepsilon_{k+1})$. Then the backwards sampling writes

- e_s^r is sampled to be v with probability

$$\frac{\alpha^v(n, m)}{\alpha^M(n, m) + \alpha^{IX}(n, m) + \alpha^{IY}(n, m)},$$

- For $k < s$ and $Z_{k+1} = (i, j)$, $i > 1, j > 1$, if $e_{k+1}^r = M$, then e_k^r is sampled to be M with probability

$$\frac{\alpha^M(i-1, j-1) \pi_{MM} \tilde{h}(X_i, Y_j | X_{i-1}, Y_{j-1})}{\alpha^M(i, j)},$$

and to be $u = I_X$ or I_Y with probability

$$\frac{\alpha^u(i-1, j-1)\pi_{uM}h(X_i, Y_j)}{\alpha^M(i, j)}.$$

Note that these latter probabilities are correctly renormalized according to Equation (3).

- For $k < s$ and $Z_{k+1} = (i, j)$, $i > 1, j > 1$, if $e_{k+1}^r = I_X$ then e_k^r is sampled to be u with probability

$$\frac{\alpha^u(i-1, j)\pi_{uI_X}f(X_i)}{\alpha^{I_X}(i, j)}, \quad (6)$$

- For $k < s$ and $Z_{k+1} = (i, j)$, $i > 1, j > 1$, if $e_{k+1}^r = I_Y$ then e_k^r is sampled to be u with probability

$$\frac{\alpha^u(i, j-1)\pi_{uI_Y}g(Y_j)}{\alpha^{I_Y}(i, j)}, \quad (7)$$

- For $k < s$ and $Z_{k+1} = (1, j)$, $j > 1$, if $e_{k+1}^r = M$ or I_X then all the values $e_l^r, 1 \leq l \leq k$ are fixed to I_Y , otherwise e_k^r is sampled to be u with probability given by (7).
- For $k < s$ and $Z_{k+1} = (i, 1)$, $i > 1$, if $e_{k+1}^r = M$ or I_Y then all the values $e_l^r, 1 \leq l \leq k$ are fixed to I_X , otherwise e_k^r is sampled to be u with probability given by (6).

According to Equation (5), we thus have sampled paths e^r satisfying the following property.

PROPOSITION 1: The paths $e \in \mathcal{E}_{n,m}$ sampled from the above scheme with parameter value θ occur with probability $\mathbb{P}_\theta(\varepsilon = e | X_{1:n}, Y_{1:m}, L_{nm} = |e|)$.

We now consider the complete log-likelihood obtained from the observed sequences and one simulated path. We have

$$\begin{aligned} \log \mathbb{P}_\theta(X_{1:n}, Y_{1:m}, \varepsilon = e^r) &= \sum_{u \in \mathcal{E}} 1_{e_1^r=u} \log \pi_u + \sum_{k=2}^{|e^r|} \sum_{u,v \in \mathcal{E}^2} 1_{e_{k-1}^r=u, e_k^r=v} \log \pi_{uv} \\ &+ \sum_{k=1}^{|e^r|} \sum_{a \in \mathcal{A}} [1_{e_k^r=I_X, X_{n_k}=a} \log f(a) + 1_{e_k^r=I_Y, Y_{m_k}=a} \log g(a)] + \sum_{k=1}^{|e^r|} \sum_{a,b \in \mathcal{A}^2} 1_{e_k^r=M, X_{n_k}=a, Y_{m_k}=b} \log h(a, b) \\ &+ \sum_{k=2}^{|e^r|} \sum_{a,b,c,d \in \mathcal{A}^4} 1_{e_k^r=e_{k-1}^r=M, (X_{n_k}, Y_{m_k}, X_{n_{k-1}}, Y_{m_{k-1}})=(a,b,c,d)} \log \tilde{h}(a, b|c, d), \end{aligned}$$

where we recall that $(n_k, m_k) = \sum_{i=1}^k e_i^r$ (for simplicity, we omit the subscript r). Then in the maximization step of **sem**, the components of θ are updated to the values: $\forall u, v \in \mathcal{E}, a, b, c, d \in \mathcal{A}$,

$$\begin{aligned} \pi_u^{(r)} &= \frac{N_{e^r}(u)}{\sum_{v \in \mathcal{E}} N_{e^r}(v)}, \quad \pi_{u,v}^{(r)} = \frac{N_{e^r}(u, v)}{\sum_{w \in \mathcal{E}} N_{e^r}(u, w)}, \quad f^{(r)}(a) = \frac{N_{e^r}(a|I_X)}{\sum_{a' \in \mathcal{A}} N_{e^r}(a'|I_X)}, \quad g^{(r)}(a) = \frac{N_{e^r}(a|I_Y)}{\sum_{a' \in \mathcal{A}} N_{e^r}(a'|I_Y)}, \\ h^{(r)}(a, b) &= \frac{N_{e^r}(a, b|M)}{\sum_{a', b' \in \mathcal{A}} N_{e^r}(a', b'|M)}, \quad \tilde{h}^{(r)}(a, b|c, d) = \frac{N_{e^r}(a, b|MM; c, d)}{\sum_{a', b' \in \mathcal{A}} N_{e^r}(a', b'|MM; c, d)}, \end{aligned} \quad (8)$$

where we use obvious notations for the counts N_e in sequence e . Namely for any $u, v \in \mathcal{E}$, and any $a, b, c, d \in \mathcal{A}$, we let $N_e(u) = \sum_k 1\{e_k = u\}$, $N_e(u, v) = \sum 1\{e_{k-1} = u, e_k = v\}$, $N_e(a|I_X) = \sum 1\{e_k = I_X, X_{n_k} = a\}$ and similarly for $N_e(a|I_Y)$, $N_e(a, b|M)$ and also $N_e(a, b|MM; c, d) = \sum 1\{e_k = e_{k-1} = M, (X_{n_k}, Y_{m_k}, X_{n_{k-1}}, Y_{m_{k-1}}) = (a, b, c, d)\}$.

In the maximization step of **saem** we have to take into account the $m(r)$ hidden paths and the stochastic approximation of $Q_r(\theta)$. Then the values of θ are updated as in (8) by replacing the counts $N_{e^r}(\cdot)$ by their counterparts $\tilde{N}_r(\cdot)$, where $\tilde{N}_r = \tilde{N}_{r-1} + \gamma_r \left(\frac{1}{m(r)} \sum_{j=1}^{m(r)} N_{e^r(j)} - \tilde{N}_{r-1} \right)$.

From a theoretical point of view, the convergence properties of the two algorithms are different. In **saem**, the sequence $\{\theta^r\}_{r \geq 1}$ converges, under general conditions, to a local maximum of the log-likelihood. It is important to note that the stochastic perturbation introduced with respect to the original **em** algorithm allows avoiding saddle points, which are possible attractive stationary points of the sequence generated by **em** (Delyon et al., 1999). In **sem**, the sequence $\{\theta^r\}_{r \geq 1}$ does not converge pointwise, but it is an homogeneous Markov chain which is ergodic under appropriate conditions, see for instance Diebolt and Ip (1996).

In this context, maximum likelihood estimators are sensitive to overfitting if there is insufficient data. Indeed, if an event has never been observed in the sequences, the corresponding estimator is not well defined, as both the numerator and denominator in (8) are zero. Pseudo-counts (Durbin et al., 1998, Chapter 3) or other smoothing techniques such as the one proposed in Kneser and Ney (1995) may be used to solve this problem. However, when using **saem** algorithm to perform maximum likelihood estimation, this problem is minimized and no smoothing strategies are required. Indeed, since at iteration r we

Table 1
Parameter values in the two data sets.

Data set 1	$\alpha = 0.4$	$\beta = 0.2$	$\gamma = 0.06$	$\lambda = 0.04$
Data set 2	$\alpha = 0.5$	$\beta = 0.15$	$\gamma = 0.05$	$\lambda = 0.02$

generate $m(r)$ hidden paths, the counts \tilde{N}_r (computed as an average over those alignments) are rarely equal to 0, even during the first iterations when γ_r is typically set to 1. However, if we want to completely avoid this possibility, we may replace event counts by pseudo-counts in the case of unobserved events, during these first iterations. Note also that in practice we do not generate whole alignments at each iteration step, but only sub-alignments of a certain given length within the current alignment. This strategy combines an MCMC procedure within **saem** and is known to preserve convergence properties of **saem** algorithm (see Arribas-Gil et al., 2009, for further details). In terms of event counts, this strategy implies that a current low estimate of an event occurrence probability will have some impact on next iteration estimate. Nonetheless, at the last iterations of the algorithm, when typically $\gamma_r < 1$, current counts are averaged with counts from previous iterations, which provides a smoothing procedure by itself.

5. Simulations

For these simulations, we consider nucleotide sequences with alphabet $\mathcal{A} = \{A, C, G, T\}$ and a constrained parameter set, motivated by the choice of an underlying evolutionary model on these sequences. As for the generation of the hidden Markov chain, we consider the TKF91 indel model with equal value for insertion and deletion rate λ (see Arribas-Gil, 2007, for more details). The transition matrix of the hidden path is thus given by

$$\pi = \frac{1}{1+\lambda} \begin{pmatrix} e^{-\lambda} & 1-e^{-\lambda} & \lambda \\ \frac{\lambda e^{-\lambda}}{1-e^{-\lambda}} & \lambda & 1+\lambda - \frac{\lambda}{1-e^{-\lambda}} \\ e^{-\lambda} & 1-e^{-\lambda} & \lambda \end{pmatrix},$$

where the order of the states in the matrix is M , I_X and I_Y . Then, we describe the conditional distribution of the sequences. For any $x \in \mathcal{A}$, we set $f(x) = g(x) = \mu_x$, where μ is going to be the stationary distribution of the substitution process, according to the parametrization of h and \tilde{h} considered below. We set the distribution h according to a simple substitution process with equal substitution rate γ and different nucleotides frequencies (modified Jukes Cantor). Let

$$h(x, y) = \begin{cases} \mu_x(1-e^{-\gamma})\mu_y & \text{if } x \neq y \\ \mu_x[\mu_x(1-e^{-\gamma}) + e^{-\gamma}] & \text{if } x = y. \end{cases}$$

The parameter \tilde{h} accounting for the context in the substitution process differs from h only in a \mathcal{C} match context. In this way, we want to take into account a possibly higher substitution rate from the pair CpG to CpA. More precisely, we let

$$\mathbb{P}(X_{N_s} = x, Y_{M_s} = y | e_s = e_{s-1} = M, X_{N_{s-1}} = x', Y_{M_{s-1}} = y') = \begin{cases} h_C(x, y) & \text{if } x' = y' = C, \\ h(x, y) & \text{otherwise.} \end{cases}$$

In a \mathcal{C} match context, we use a model initially introduced in two slightly different forms by Felsenstein (1984) and Hasegawa et al. (1985). This model considers different substitution rates for transitions (*i.e.* substitutions within the chemical class of purines $\mathcal{R} = \{A, G\}$ or pyrimidines $\mathcal{Y} = \{C, T\}$) and for transversions (substitutions modifying the chemical class) as well as different nucleotide frequencies μ . Denoting by \bar{x} the other nucleotide in the same chemical class as x (namely $x \neq \bar{x}$ and either $\{x, \bar{x}\} \in \mathcal{R}$ or $\{x, \bar{x}\} \in \mathcal{Y}$), we get

$$h_C(x, y) = \begin{cases} \mu_x e^{-(\alpha+\beta)} + \mu_x e^{-\beta}(1-e^{-\alpha}) \frac{\mu_x}{\mu_x + \mu_{\bar{x}}} + \mu_x(1-e^{-\beta})\mu_x & \text{if } y = x, \\ \mu_x e^{-\beta}(1-e^{-\alpha}) \frac{\mu_{\bar{x}}}{\mu_{\bar{x}} + \mu_x} + \mu_x(1-e^{-\beta})\mu_{\bar{x}} & \text{if } y = \bar{x}, \\ \mu_x(1-e^{-\beta})\mu_y & \text{otherwise.} \end{cases}$$

We refer to Felsenstein (2003) for more details on substitution processes.

According to this parametrization we have simulated two sets of data with an alignment length of 2000 base pairs (bp) and parameter values described in Table 1. In these data sets, the substitution rate γ is larger than the indel rate λ as expected in biological sequences, and the transition and transversion substitution rates in a \mathcal{C} match context (α and β) are larger than the regular substitution rate γ . In both sets of data we have set the stationary distribution of the substitution process to

$$\mu_A = 0.225 \quad \mu_C = 0.275 \quad \mu_G = 0.275 \quad \mu_T = 0.225,$$

in order to obtain a GC content (proportion of Gs or Cs) larger than 50%.

For each one of the data sets we have conducted two estimation procedures. On the one hand, we have estimated the whole parameter $(\pi, f, g, h, \tilde{h})$ via the **saem** algorithm as explained in Section 4.2. On the other hand, we have combined the **saem** algorithm with numerical optimization to estimate only the evolutionary parameters λ , γ , α and β . The first procedure

has the advantages of being robust against misspecification of the underlying evolutionary model and relying on an explicit maximization step based on the counts of events. The second procedure is computationally more expensive, however, it is more parsimonious and it provides a straightforward evolutionary interpretation of the parameter values. In both cases we have performed 150 iterations of the **saem** algorithm with the parameter γ_r of the stochastic approximation set to 1 for $r = 1, \dots, 100$ and to $1/(r - 100)$ for $r = 101, \dots, 150$. The number of simulated hidden paths is $m(r) = 5$, $r = 1, \dots, 20$ and $m(r) = 10$, $r = 21, \dots, 150$. We used the same initial values of the parameters for every simulation run over the two simulated data sets. These initial values, estimates and standard deviations are given in Tables 2, 3 (first procedure) and 4 (second procedure). In Figure 2 we present the distribution of estimates obtained with the second estimation approach for the two simulated data sets. We can observe that both procedures provide unbiased and very accurate estimates. In Figure 3 we show the convergence of the **saem** estimates on a single simulation run from the second data set. We use a logarithmic scale for the x -axis to highlight the very fast convergence of the parameters to their true values.

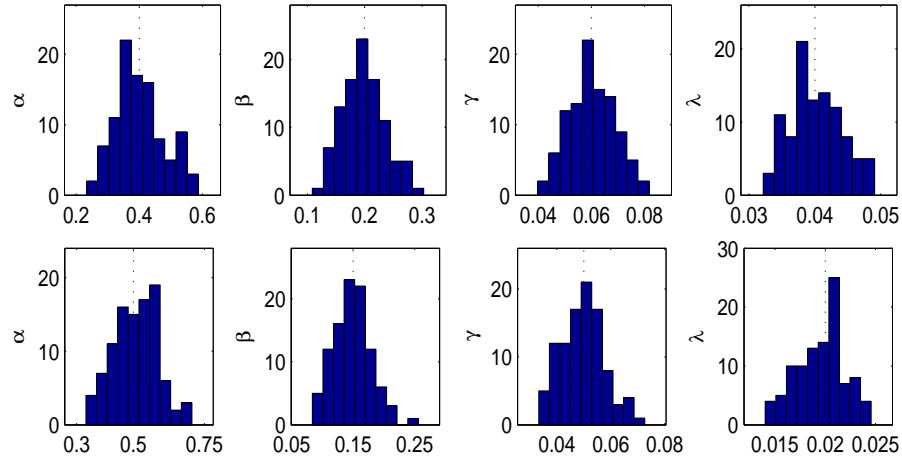


Figure 2. Distribution of **saem** estimates of the reduced parameter vector $(\alpha, \beta, \gamma, \lambda)$ over 100 simulation runs for the two simulated data sets (Data set 1 on top, Data set 2 on bottom). The real values of the parameters are displayed in dotted line.

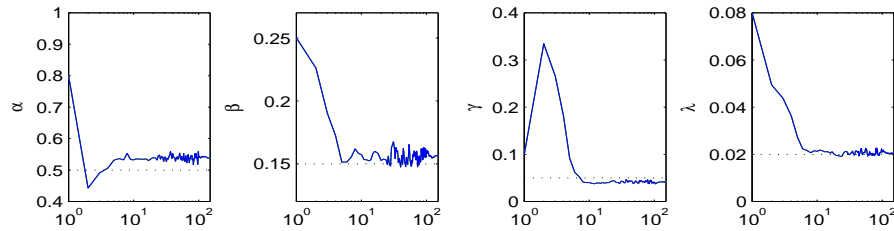


Figure 3. **saem** estimates of the reduced parameter vector $(\alpha, \beta, \gamma, \lambda)$ over iterations in a single simulation run from the second data set. A logarithmic scale is used for the x -axis. The true parameter values are displayed in dotted line.

6. Application to real data

In this section we illustrate our method through an alignment of the human alpha-globin pseudogene (HBPA1) with its functional counterpart, the human alpha-globin gene (HBA1). This example is inspired by the work of Bulmer (1986), who studied the neighbouring base effects on substitution rates in a series of vertebrate pseudogenes, including the human alpha-globin, concluding that there is an increase in the frequency of substitutions from the dinucleotide CpG.

We have extracted the sequences of the gene and the pseudogene, who are located in the human alpha-globin gene cluster on chromosome 16, from the UCSC Genome Browser database (Fujita et al., 2010). We have considered the whole sequences (coding and non-coding sequence in the case of the functional gene) for the alignment, obtaining sequences lengths of 842 bp for HBA1 and 812 bp for HBPA1. Because of the presence of introns and exons in the HBA1 sequence, it is natural to consider a model allowing for two different substitution behaviors along the sequence. That is why we have used a model for which the state space of the hidden Markov chain is $\{M_1, M_2, I_X, I_Y\}$, where M_i , $i = 1, 2$, stand for two different match states, with different associated substitution processes h_i, \tilde{h}_i , $i = 1, 2$. This kind of model may also handle fragments insertion and deletion

Table 2

Mean values and standard deviations of estimates over 100 simulation runs for Data Set 1. The estimation has been performed without taking into account the specific parametrization used to generate the data.

	Initial value	True value	Estimate (sd)
π_{MM}	0.85	0.9238	0.9184 (0.0101)
π_{MI_X}	0.075	0.0377	0.0407 (0.0074)
π_{MI_Y}	0.075	0.0385	0.0410 (0.0077)
$\pi_{I_X M}$	0.85	0.9424	0.8807 (0.0998)
$\pi_{I_X I_X}$	0.075	0.0385	0.0387 (0.0308)
$\pi_{I_X I_Y}$	0.075	0.0191	0.0805 (0.1034)
$\pi_{I_Y M}$	0.85	0.9238	0.8788 (0.0982)
$\pi_{I_Y I_X}$	0.075	0.0377	0.0821 (0.0987)
$\pi_{I_Y I_Y}$	0.075	0.0385	0.0391 (0.0305)
$h(A, A)$	0.0625	0.2148	0.2155 (0.0109)
$h(A, C)$	0.0625	0.0036	0.0031 (0.0019)
$h(A, G)$	0.0625	0.0036	0.0032 (0.0019)
$h(A, T)$	0.0625	0.0029	0.0027 (0.0018)
$h(C, A)$	0.0625	0.0036	0.0034 (0.0021)
$h(C, C)$	0.0625	0.2634	0.2628 (0.0130)
$h(C, G)$	0.0625	0.0044	0.0044 (0.0024)
$h(C, T)$	0.0625	0.0036	0.0034 (0.0021)
$h(G, A)$	0.0625	0.0036	0.0030 (0.0017)
$h(G, C)$	0.0625	0.0044	0.0041 (0.0023)
$h(G, G)$	0.0625	0.2634	0.2653 (0.0120)
$h(G, T)$	0.0625	0.0036	0.0032 (0.0018)
$h(T, A)$	0.0625	0.0029	0.0026 (0.0016)
$h(T, C)$	0.0625	0.0036	0.0031 (0.0017)
$h(T, G)$	0.0625	0.0036	0.0035 (0.0021)
$h(T, T)$	0.0625	0.2148	0.2166 (0.0113)
$h_C(A, A)$	0.0625	0.1600	0.1626 (0.0169)
$h_C(A, C)$	0.0625	0.0112	0.0108 (0.0053)
$h_C(A, G)$	0.0625	0.0446	0.0454 (0.0111)
$h_C(A, T)$	0.0625	0.0092	0.0091 (0.0051)
$h_C(C, A)$	0.0625	0.0112	0.0102 (0.0056)
$h_C(C, C)$	0.0625	0.2055	0.2072 (0.0193)
$h_C(C, G)$	0.0625	0.0137	0.0126 (0.0056)
$h_C(C, T)$	0.0625	0.0446	0.0429 (0.0108)
$h_C(G, A)$	0.0625	0.0446	0.0440 (0.0098)
$h_C(G, C)$	0.0625	0.0137	0.0140 (0.0061)
$h_C(G, G)$	0.0625	0.2055	0.2050 (0.0174)
$h_C(G, T)$	0.0625	0.0112	0.0110 (0.0053)
$h_C(T, A)$	0.0625	0.0092	0.0086 (0.0047)
$h_C(T, C)$	0.0625	0.0446	0.0456 (0.0106)
$h_C(T, G)$	0.0625	0.0112	0.0111 (0.0062)
$h_C(T, T)$	0.0625	0.1600	0.1598 (0.0197)

in contrast to single nucleotide indels (see Arribas-Gil et al., 2009, for more details). As in the simulation studies of the previous section, \tilde{h} is the substitution matrix in a \mathcal{C} match context, and h is the substitution matrix in any other case. In this way, we want to take into account a possibly higher transition rate from nucleotide G occurring in the dinucleotide CpG. In order to avoid any possible misspecification of the underlying indel and substitution processes, we have decided not to parametrize any of those, conducting an estimation approach equivalent to the first procedure described in the previous section. The only difference here lies in the dimension of the model parameter, since this parameter is now $(\pi, f, g, h_1, h_2, \tilde{h}_1, \tilde{h}_2)$, where π is a 4×4 transition probability matrix, and $h_i, \tilde{h}_i, i = 1, 2$ are four different 4×4 stochastic vectors, yielding a total number of $12 + 3 + 3 + 4 \times 15 = 78$ free parameters.

We have run **saem** algorithm on the sequences, performing 500 iterations with γ_r set to 1 for $r = 1, \dots, 400$ and to $1/(r - 400)$

Table 3

Mean values and standard deviations of estimates over 100 simulation runs for Data set 2. The estimation has been performed without taking into account the specific parametrization used to generate the data.

	Initial value	True value	Estimate (sd)
π_{MM}	0.85	0.9610	0.9559 (0.0071)
π_{MI_X}	0.075	0.0194	0.0221 (0.0051)
π_{MI_Y}	0.075	0.0196	0.0219 (0.0055)
$\pi_{I_X M}$	0.85	0.9706	0.8477 (0.1262)
$\pi_{I_X I_X}$	0.075	0.0196	0.0199 (0.0221)
$\pi_{I_X I_Y}$	0.075	0.0098	0.1323 (0.1289)
$\pi_{I_Y M}$	0.85	0.9610	0.8407 (0.1321)
$\pi_{I_Y I_X}$	0.075	0.0194	0.1367 (0.1379)
$\pi_{I_Y I_Y}$	0.075	0.0196	0.0226 (0.0252)
$h(A, A)$	0.0625	0.2165	0.2168 (0.0114)
$h(A, C)$	0.0625	0.0030	0.0027 (0.0015)
$h(A, G)$	0.0625	0.0030	0.0027 (0.0014)
$h(A, T)$	0.0625	0.0025	0.0024 (0.0015)
$h(C, A)$	0.0625	0.0030	0.0026 (0.0017)
$h(C, C)$	0.0625	0.2653	0.2673 (0.0122)
$h(C, G)$	0.0625	0.0037	0.0033 (0.0020)
$h(C, T)$	0.0625	0.0030	0.0027 (0.0016)
$h(G, A)$	0.0625	0.0030	0.0023 (0.0015)
$h(G, C)$	0.0625	0.0037	0.0033 (0.0020)
$h(G, G)$	0.0625	0.2653	0.2643 (0.0096)
$h(G, T)$	0.0625	0.0030	0.0023 (0.0013)
$h(T, A)$	0.0625	0.0025	0.0024 (0.0015)
$h(T, C)$	0.0625	0.0030	0.0025 (0.0016)
$h(T, G)$	0.0625	0.0030	0.0027 (0.0015)
$h(T, T)$	0.0625	0.2165	0.2196 (0.0102)
$h_C(A, A)$	0.0625	0.1588	0.1595 (0.0159)
$h_C(A, C)$	0.0625	0.0086	0.0069 (0.0038)
$h_C(A, G)$	0.0625	0.0505	0.0504 (0.0096)
$h_C(A, T)$	0.0625	0.0071	0.0067 (0.0042)
$h_C(C, A)$	0.0625	0.0086	0.0087 (0.0045)
$h_C(C, C)$	0.0625	0.2053	0.2053 (0.0199)
$h_C(C, G)$	0.0625	0.0105	0.0103 (0.0046)
$h_C(C, T)$	0.0625	0.0505	0.0504 (0.0101)
$h_C(G, A)$	0.0625	0.0505	0.0515 (0.0102)
$h_C(G, C)$	0.0625	0.0105	0.0097 (0.0052)
$h_C(G, G)$	0.0625	0.2053	0.2036 (0.0191)
$h_C(G, T)$	0.0625	0.0086	0.0078 (0.0043)
$h_C(T, A)$	0.0625	0.0071	0.0078 (0.0041)
$h_C(T, C)$	0.0625	0.0505	0.0515 (0.0098)
$h_C(T, G)$	0.0625	0.0086	0.0084 (0.0040)
$h_C(T, T)$	0.0625	0.1588	0.1613 (0.0166)

for $r = 401, \dots, 500$. The number of simulated hidden paths is $m(r) = 5$, $r = 1, \dots, 20$ and $m(r) = 10$, $r = 21, \dots, 500$. In Table 5, we present the estimated substitution probabilities from nucleotide G. As expected, we found an increase in the substitution occurrence for G in a \mathcal{C} match context. We present in Figure 4 the posterior probability distribution over the set of possible alignment columns, match, insertion and deletion (the two match states have been merged) for every pair of aligned positions in a consensus alignment obtained at the convergence stage (last iterations) of **saem** algorithm.

In order to compare our method to a traditional pair-HMM alignment, we have also run **saem** algorithm, with the same settings as before, on a simpler model without substitution context-dependence. In this model we consider again two different substitution regimes, so the parameter vector is now (π, f, g, h_1, h_2) , yielding a total number of $12 + 3 + 3 + 2 \times 15 = 48$ free parameters. We present in Figure 5 the posterior probability distribution over the set of possible alignment columns, match,

Table 4

Mean values and standard deviations of estimates over 100 simulation runs for the two data sets. The estimation has been performed by numerical optimization on the reduced parameter vector $(\alpha, \beta, \gamma, \lambda)$.

Data set 1			
	Initial value	True value	Estimate (sd)
α	0.8	0.4	0.400 (0.0750)
β	0.25	0.2	0.1985 (0.0372)
γ	0.1	0.06	0.0602 (0.0087)
λ	0.08	0.04	0.0402 (0.0040)

Data set 2			
	Initial value	True value	Estimate (sd)
α	0.8	0.5	0.5091 (0.0772)
β	0.25	0.15	0.1493 (0.0296)
γ	0.1	0.05	0.0499 (0.0079)
λ	0.08	0.02	0.0196 (0.0024)

Table 5

Relative substitution probabilities for G , namely $\phi_i(G, \cdot) / \sum_{a \in \mathcal{A}} \phi_i(G, a)$, $i = 1, 2$, $\phi = h, \tilde{h}$. The probability of transition for G increases from 0.08 in the general context to 0.88 in the \tilde{C} match context in the first substitution regime, and from 0.02 to 0.15 in the second.

	A	C	G	T
h_1	0.0839	0.0759	0.8398	0.0004
\tilde{h}_1	0.8835	0.0093	0.0951	0.0121
h_2	0.0242	0.0596	0.8681	0.0481
\tilde{h}_2	0.1508	0.1420	0.7058	0.0014

insertion or deletion (the two match states have been merged) for every pair of aligned positions in a consensus alignment obtained at the convergence stage (last iterations) of **saem** algorithm. This posterior probability is quite similar to the one obtained with the context-dependent model.

In order to further compare the two models, we have calculated a BIC-type criterion penalizing the maximum likelihood by the number of parameters. Specifically, we have computed $BIC = -2L + k \log(n)$, where L is the value of the log-likelihood at $\hat{\theta}_{ML}$, n is the sample size and k is the total number of free parameters. We have taken $n = 842$, the length of the longest sequence, since the length of the alignment is unknown, but is equivalent to those of the observed sequences. Indexing by 1 the context-dependent model and by 2 the non context-dependent one, we get

$$BIC_1 = -2 \cdot (-7128.50) + 78 \log(842) = 14782.39$$

$$BIC_2 = -2 \cdot (-7337.40) + 48 \log(842) = 14998.12,$$

which means that the context-dependent model better fits this data set.

7. Discussion

We have proposed a model for statistical sequence alignment accounting for context-dependence in the substitution process. This model extends the pair hidden Markov model in the sense that, conditional on the alignment, the presence of particular nucleotides at the different sites of a sequence are no longer independent. We have adapted the traditional dynamic programming algorithms to the new framework, and we have proposed an efficient estimation method based on **saem** algorithm. We have obtained asymptotic results for maximum likelihood estimation on the new model, and through a simulation study, we have shown the accuracy of our algorithms on finite sample estimation. Finally, we have illustrated our methods with the alignment of the human alpha-globin pseudogene and its functional counterpart. We have compared the new model with a classical pair-

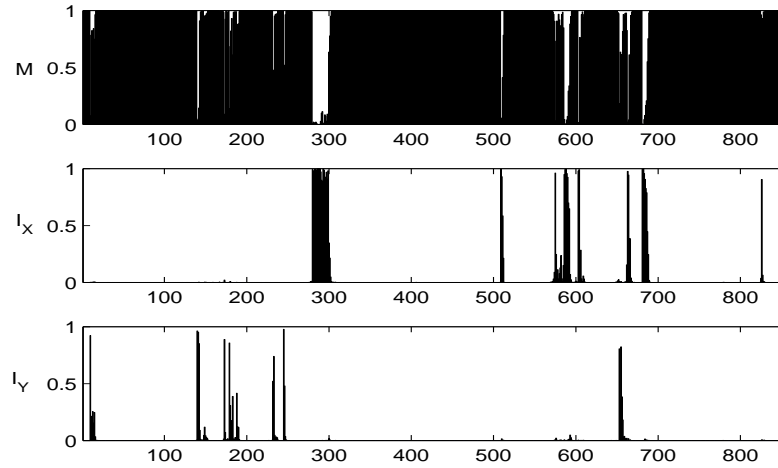


Figure 4. Posterior probability distribution of the alignment states at the maximum likelihood parameter estimate with the substitution context-dependent model.

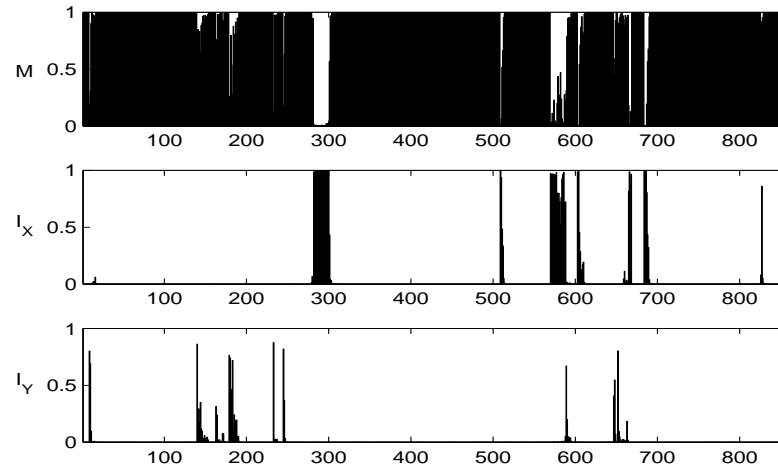


Figure 5. Posterior probability distribution of the alignment states at the maximum likelihood parameter estimate without taking into account possible substitution context-dependence.

HMM through a model selection criterion, concluding that taking into account the context-dependence on the substitution process improves the fitting of the data.

References

- Arribas-Gil, A. (2007). *Estimation dans des modèles à variables cachées : alignement de séquences biologiques et modèles d'évolution*. PhD thesis, Université Paris-Sud, France. http://halweb.uc3m.es/esp/Personal/personas/aarribas/esp/docs/these_arribas_gil.pdf.
- Arribas-Gil, A., Gassiat, E., and Matias, C. (2006). Parameter estimation in pair-hidden Markov models. *Scand J Stat* **33**, 651–671.
- Arribas-Gil, A., Metzler, D., and Plouhinec, J.-L. (2009). Statistical alignment with a sequence evolution model allowing rate heterogeneity along the sequence. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **6**, 281–295.
- Baum, L., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann Math Stat* **41**, 164–171.
- Bulmer, M. (1986). Neighboring base effects on substitution rates in pseudogenes. *Mol Biol Evol* **3**, 322–329.

- Celeux, G. and Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly* **2**, 73–82.
- Delyon, B., Lavielle, M., and Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *Ann Stat* **27**, 94–128.
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J Roy Stat Soc B* **39**, 1–38.
- Dewey, C. N., Huggins, P. M., Woods, K., Sturmfels, B., and Pachter, L. (2006). Parametric alignment of *Drosophila* genomes. *PLoS Comput Biol* **2**, e73.
- Diebolt, J. and Celeux, G. (1993). Asymptotic properties of a stochastic EM algorithm for estimating mixing proportions. *Comm Statist. Stochastic Models* **9**, 599–613.
- Diebolt, J. and Ip, E. H. S. (1996). A stochastic EM algorithm for approximating the maximum likelihood estimate. In W. R. Gilks, S. T. Richardson, D. J. S., editor, *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London.
- Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK.
- Felsenstein, J. (1984). Distance methods for inferring phylogenies: A justification. *Evolution* **38**, pp. 16–24.
- Felsenstein, J. (2003). *Inferring phylogenies*. Sinauer Associates.
- Fujita, P. A., Rhead, B., and Zweig, A. A. *et al.* (2010). The UCSC Genome Browser database: update 2011. *Nucleic Acids Res.*
- Gambin, A., Tiuryn, J., and Tyszkiewicz, J. (2006). Alignment with context dependent scoring function. *J Comput Biol* **13**, 81–101.
- Gambin, A. and Wojtalewicz, P. (2007). CTX-BLAST: context sensitive version of protein BLAST. *Bioinformatics* **23**, 1686–1688.
- Hasegawa, M., Kishino, H., and Yano, T.-A. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol* **22**, 160–174.
- Hickey, G. and Blanchette, M. (2011a). A probabilistic model for sequence alignment with context-sensitive indels. In Bafna, V. and Sahinalp, S., editors, *Research in Computational Molecular Biology*, volume 6577 of *Lecture Notes in Computer Science*, pages 85–103. Springer Berlin / Heidelberg.
- Hickey, G. and Blanchette, M. (2011b). A probabilistic model for sequence alignment with context-sensitive indels. *J Comput Biol.* to appear.
- Hobolth, A. (2008). A Markov chain Monte Carlo Expectation Maximization algorithm for statistical analysis of DNA sequence evolution with neighbor-dependent substitution rates. *Journal of Computational and Graphical Statistics* **17**, 138–162.
- Holmes, I. (2005). Using evolutionary Expectation Maximization to estimate indel rates. *Bioinformatics* **21**, 2294–2300.
- Holmes, I. and Durbin, R. (1998). Dynamic programming alignment accuracy. *J Comput Biol.* **5**, 493–504.
- Huang, X. (1994). A context dependent method for comparing sequences. In *Combinatorial pattern matching (Asilomar, CA, 1994)*, volume 807 of *Lecture Notes in Comput. Sci.*, pages 54–63. Springer, Berlin.
- Jukes, T. H. and Cantor, C. R. (1969). Evolution of protein molecules. In Munro, H., editor, *Mammalian Protein Metabolism III*, pages 21–132. Academic Press, New York.
- Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184.
- Knudsen, B. and Miyamoto, M. (2003). Sequence alignments and pair hidden Markov models using evolutionary history. *J Mol Biol* **333**, 453–460.
- Lunter, G., Drummond, A. J., Miklós, I., and Hein, J. (2005). Statistical alignment: Recent progress, new applications, and challenges. In Nielsen, R., editor, *Statistical Methods in Molecular Evolution*, Statistics for Biology and Health, pages 375–405. Springer New York.
- Lunter, G., Rocco, A., Mimouni, N., Heger, A., Caldeira, A., and Hein, J. (2007). Uncertainty in homology inferences: Assessing and improving genomic sequence alignment. *Genome Res* **18**, 000.
- Metzler, D. (2003). Statistical alignment based on fragment insertion and deletion models. *Bioinformatics* **19**, 490–499.
- Miklós, I. (2003). Algorithm for statistical alignment of two sequences derived from a poisson sequence length distribution. *Discrete Appl Math* **127**, 79 – 84. Computational Molecular Biology Series - Issue IV.
- Miklós, I., Lunter, G., and Holmes, I. (2004). A ‘Long Indel’ model for evolutionary sequence alignment. *Mol Biol Evol* **21**, 529–540.
- Miklós, I. and Toroczka, Z. (2001). An improved model for statistical alignment. In Gascuel, O. and Moret, B., editors, *Algorithms in Bioinformatics*, volume 2149 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin / Heidelberg.
- Siepel, A. and Haussler, D. (2004). Computational identification of evolutionarily conserved exons. In *Proceedings of the eighth annual international conference on Research in computational molecular biology, RECOMB ’04*, pages 177–186, New York, NY, USA. ACM.
- Thorne, J., Kishino, H., and Felsenstein, J. (1991). An evolutionary model for maximum likelihood alignment of DNA sequences. *J Mol Evol* **33**, 114–124.

- Thorne, J., Kishino, H., and Felsenstein, J. (1992). Inching toward reality: an improved likelihood model of sequence evolution. *J Mol Evol* **34**, 3–16.
- van der Vaart, A. W. (1998). *Asymptotic statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge.
- Varela, M. and Amos, W. (2009). Evidence for nonindependent evolution of adjacent microsatellites in the human genome. *J Mol Evol* **68**, 160–170.
- Wald, A. (1949). Note on the consistency of the maximum likelihood estimate. *Ann. Math. Statistics* **20**, 595–601.
- Waterman, M. S. and Eggert, M. (1987). A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *J Mol Biol* **197**, 723 – 728.
- Wong, K. M., Suchard, M. A., and Huelsenbeck, J. P. (2008). Alignment uncertainty and genomic analysis. *Science* **319**, 473–476.

Appendix

The need for a stochastic approximation

Let us explain why **em** algorithm may not be applied in pair-HMM. Denoting by L_{nm} the random value $s \geq 1$ such that $Z_s = (n, m)$ (the first and only hitting time for the point (n, m) , which is not necessarily finite), the complete log-likelihood writes

$$\begin{aligned} \log \mathbb{P}_\theta(X_{1:n}, Y_{1:m}, L_{nm}, \varepsilon_{1:L_{nm}}) &= \sum_{s=n \vee m}^{n+m} 1_{L_{nm}=s} \left\{ \sum_{u \in \mathcal{E}} 1_{\varepsilon_1=u} \log \pi_u + \sum_{k=2}^s \sum_{u,v \in \mathcal{E}^2} 1_{\varepsilon_{k-1}=u, \varepsilon_k=v} \log \pi_{uv} \right. \\ &+ \sum_{k=1}^s \sum_{a \in \mathcal{A}} [1_{\varepsilon_k=I_X, X_{N_k}=a} \log f(a) + 1_{\varepsilon_k=I_Y, Y_{M_k}=a} \log g(a)] + \sum_{k=1}^s \sum_{a,b \in \mathcal{A}^2} 1_{\varepsilon_k=M, \varepsilon_{k-1} \neq M, X_{N_k}=a, Y_{M_k}=b} \log h(a, b) \\ &\left. + \sum_{k=2}^s \sum_{a,b,c,d \in \mathcal{A}^4} 1_{\varepsilon_k=\varepsilon_{k-1}=M, (X_{N_k}, Y_{M_k}, X_{N_{k-1}}, Y_{M_{k-1}})=(a,b,c,d)} \log \tilde{h}(a, b|c, d) \right\}. \end{aligned}$$

To simplify notations, we let $\mathbf{X} := X_{1:n}$, $\mathbf{Y} := Y_{1:m}$, $L := L_{nm}$ and $\mathcal{F}_s = \{\mathbf{X}, \mathbf{Y}, L = s\}$. Moreover, we let $(\varepsilon, X, Y)_k := (\varepsilon_k, X_{N_k}, Y_{M_k})$. Taking the expectation of the complete log-likelihood, conditional on the observed sequences, under a current parameter value θ' , leads to

$$\begin{aligned} \mathbb{E}_{\theta'}(\log \mathbb{P}_\theta(\mathbf{X}, \mathbf{Y}, L, \varepsilon_{1:L}) | \mathbf{X}, \mathbf{Y}) &= \sum_{s=n \vee m}^{n+m} \mathbb{P}_{\theta'}(L = s | \mathbf{X}, \mathbf{Y}) \left\{ \sum_{u \in \mathcal{E}} \mathbb{P}_{\theta'}(\varepsilon_1 = u | \mathcal{F}_s) \log \pi_u \right. \\ &+ \sum_{k=2}^s \sum_{u,v \in \mathcal{E}^2} \mathbb{P}_{\theta'}((\varepsilon_{k-1}, \varepsilon_k) = (u, v) | \mathcal{F}_s) \log \pi_{uv} \\ &+ \sum_{k=1}^s \sum_{a \in \mathcal{A}} [\mathbb{P}_{\theta'}((\varepsilon, X)_k = (I_X, a) | \mathcal{F}_s) \log f(a) + \mathbb{P}_{\theta'}((\varepsilon, Y)_k = (I_Y, a) | \mathcal{F}_s) \log g(a)] \\ &+ \sum_{k=1}^s \sum_{a,b} \mathbb{P}_{\theta'}(\varepsilon_{k-1} \neq M, (\varepsilon, X, Y)_k = (M, a, b) | \mathcal{F}_s) \log h(a, b) \\ &\left. + \sum_{k=2}^s \sum_{a,b,c,d} \mathbb{P}_{\theta'}((\varepsilon_k, \varepsilon_{k-1}, (X, Y)_k, (X, Y)_{k-1}) = (M, M, a, b, c, d) | \mathcal{F}_s) \log \tilde{h}(a, b|c, d) \right\}. \end{aligned}$$

There are two main issues at stake here. First, the quantity $\mathbb{P}_{\theta'}(L_{nm} | X_{1:n}, Y_{1:m})$ is not given by the forward-backward equations. Second, computing this conditional expectation would also require the knowledge of the quantities

$$\mathbb{P}_{\theta'}(\varepsilon_{k-1}, \varepsilon_k | \mathbf{X}, \mathbf{Y}, L = s).$$

However, the forward backward equations rather give access to quantities as for instance

$$\alpha^{IX}(i-1, j-1) \pi_{IXM} h(X_i, Y_j) \beta^M(i, j) = \mathbb{P}_\theta(\varepsilon_{k-1} = I_X, \varepsilon_k = M | \mathbf{X}, \mathbf{Y}, Z_k = (i, j)).$$

In other words, the conditional distribution of $(\varepsilon_{k-1}, \varepsilon_k)$ is known only conditional on the extra knowledge of the position Z_k of the path in the lattice at time k . In conclusion, in pair-HMM, the forward-backward equations are not sufficient to compute the expectation of the complete log-likelihood, conditional on the observed sequences.